



PUC Minas

# Linguagens de Programação (Ciência da Computação)

Professora: M. Sc. Luciana De Nardin  
luciana@pucpcaldas.br

# **Plano de ensino**

## ◎ Ementa

- ◎ Evolução das linguagens de programação
- ◎ Conceitos de linguagens de programação
- ◎ Paradigmas de programação: imperativo, funcional, lógico e orientado a objetos
- ◎ Sistemas de tipos
- ◎ Conceitos e princípios da programação orientada a objetos
- ◎ Exemplos práticos e estudos de caso em linguagens dos diversos paradigmas

## ◎ **Objetivos**

- ◎ Contextualizar linguagens de programação e sua evolução histórica
- ◎ Conhecer os conceitos fundamentais das linguagens de programação
- ◎ Conhecer os principais paradigmas de programação (imperativas, declarativas, funcionais e orientadas a objetos)
- ◎ Experimentar o uso de diversos tipos de linguagens de programação na resolução de diferentes tipos de problemas
- ◎ Investigar os principais aspectos envolvidos no projeto de linguagens de programação



# ◎ Conteúdo programático

## ◎ Unidade I – Introdução

- 1.1 Objetivos
- 1.2 Definição de linguagem de programação
- 1.3 Breve histórico das linguagens

## ◎ Unidade II - Especificação de linguagens de programação

- 2.1 Sintaxe
- 2.2 Semântica

## ◎ **Conteúdo programático**

- ◎ Unidade III - Tradução de linguagens de programação
  - 3.1 Interpretação
  - 3.2 Compilação
- ◎ Unidade IV - Características do projeto de linguagens de programação
  - 4.1 Propriedades fundamentais
  - 4.2 Paradigmas

## ◎ Conteúdo programático

### ◎ Unidade V - Paradigmas de linguagens de programação

#### ■ 5.1 Paradigma imperativo

- 5.1.1 Estruturas de controle
- 5.1.2 Tipos de dados
- 5.1.3 Procedimentos e funções
- 5.1.4 Linguagens **C** e **Pascal**

## ◎ Conteúdo programático

### ■ 5.2 Paradigma lógico

- 5.2.1 Características da programação lógica
- 5.2.2 Axiomas e regras de inferência
- 5.2.3 Estrutura de dados
- 5.2.4 Fluxo de controle e cortes
- 5.2.5 Linguagem **Prolog**



## ◎ Conteúdo programático

### ■ 5.3 Paradigma funcional

- 5.3.1 Características da programação funcional
- 5.3.2 Funções e aplicações
- 5.3.3 Recursividade
- 5.3.4 Linguagem **Lisp**

## ◎ Conteúdo programático

- 5.4 Paradigma orientado a objetos
  - 5.4.1 Características da programação orientada a objetos
  - 5.4.2 Conceitos de orientação a objetos
  - 5.4.3 Linguagem **Java**

## ◎ Sistema de avaliação

- ◎ Três avaliações teóricas individuais (60 pontos)
- ◎ Listas de exercícios e atividades em aula individuais ou em grupo (40 pontos)
- ◎ Reavaliação individual (100 pontos)

## ◎ Bibliografia básica

- ◎ BRATKO, Ivan. *Prolog: programming for artificial intelligence*. Harlow: Addison Wesley, 1990
- ◎ DEITEL, Paul J.; Deitel, Harvey M. Java: como programar - 8ª edição
- ◎ PRATT, Terrence. *Programming Languages: design and implementation*.
- ◎ SEBESTA, R. Conceitos de Linguagens de Programação. Porto Alegre: Bookman, 2003. 5.ed.
- ◎ TANIMOTO, Steven L. *The elements of artificial intelligence using common lisp*. 2nd ed. New York: Computer Science Press, 1995



# 1

## Unidade I

## ◎ **Roteiro**

- ◎ Objetivos
- ◎ Definição de linguagens de programação
- ◎ Histórico das linguagens

## ◎ Objetivos

### ◎ Por que estudar os paradigmas das linguagens de programação?

- Programação é fundamental para computação
- Aumento da capacidade de expressar ideias
  - Difícil pessoas expressarem estruturas que não conseguem descrever → programadores, idem!!!
- Maior embasamento para escolha de linguagens apropriadas
  - Uso contínuo da linguagem que mais domina, mesmo que inadequada

## ◎ Objetivos

- ◎ Por que estudar os paradigmas das linguagens de programação?
  - Capacidade aumentada para aprender novas linguagens
    - Computação >> evolução contínua!!
    - Aprender é mais fácil para quem conhece os conceitos gerais de LP's
  - Entender melhor a importância da implementação
    - Usar a linguagem inteligentemente!



## ◎ Objetivos

- ◎ Quais os domínios de programação (áreas de atuação)?
  - Aplicações científicas (início da déc. 40)
    - Estruturas de dados simples (FORTRAN >> *FORmula TRANslator*)
  - Aplicações comerciais (déc. 50)
    - 1960: COBOL (até hoje...)
    - Produção de relatórios e armazenamento de dados
    - Consolidação de aplicações (planilhas, folhas de pagamento, etc.)

## ◎ **Objetivos**

### ◎ Quais os domínios de programação (áreas de atuação)?

- Inteligência artificial (déc. 60)
  - 1ª linguagem: LISP (1959)
  - Mais utilizada: PROLOG
- Programação de sistemas (déc. 60)
  - SO's
  - Eficiência na execução
  - Mais utilizada: C (Unix foi escrito em C)

## ◎ Objetivos

### ◎ Como avaliar uma linguagem de programação?

#### ■ Legibilidade

- Facilidade com que um programa pode ser lido
- Simplicidade (não muita – Assembly) >> legibilidade
- Multiplicidade de operadores (c++ ou  $c=c+1$  ou  $c+=1$ )
- Sobrecarga de operadores
- Ortogonalidade
- Declaração de controle >> *top down* (sem *goto's*)

## ◎ Objetivos

### ◎ Como avaliar uma linguagem de programação?

- Capacidade de escrita
  - Facilidade de usar a linguagem em um domínio específico
- Confiabilidade
  - Confiável se executa de acordo com suas especificações sob quaisquer circunstâncias
  - Checagem de tipos >> compatibilidade

Java e Pascal 😄

C 😞



## ◎ Objetivos

### ◎ Como avaliar uma linguagem de programação?

- Custo

- Custo de treinamento, escritabilidade, compilação, execução, implementação, confiabilidade e manutenção

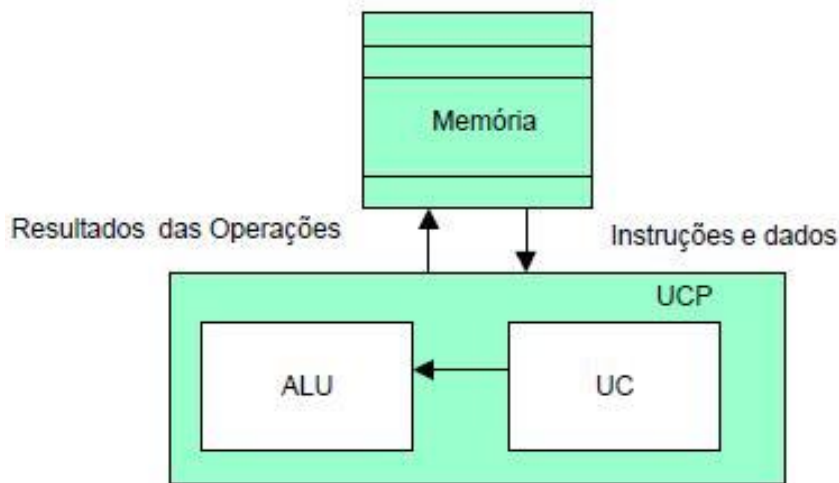
- Outros critérios

Portabilidade, generalização, boa definição,  
boas ferramentas, etc.

# Objetivos

## Influências sobre o projeto da linguagem

- Arquitetura do computador
- Máquinas de Von Neumann >> linguagens imperativas



## ◎ **Objetivos**

### ◎ Influências sobre o projeto da linguagem

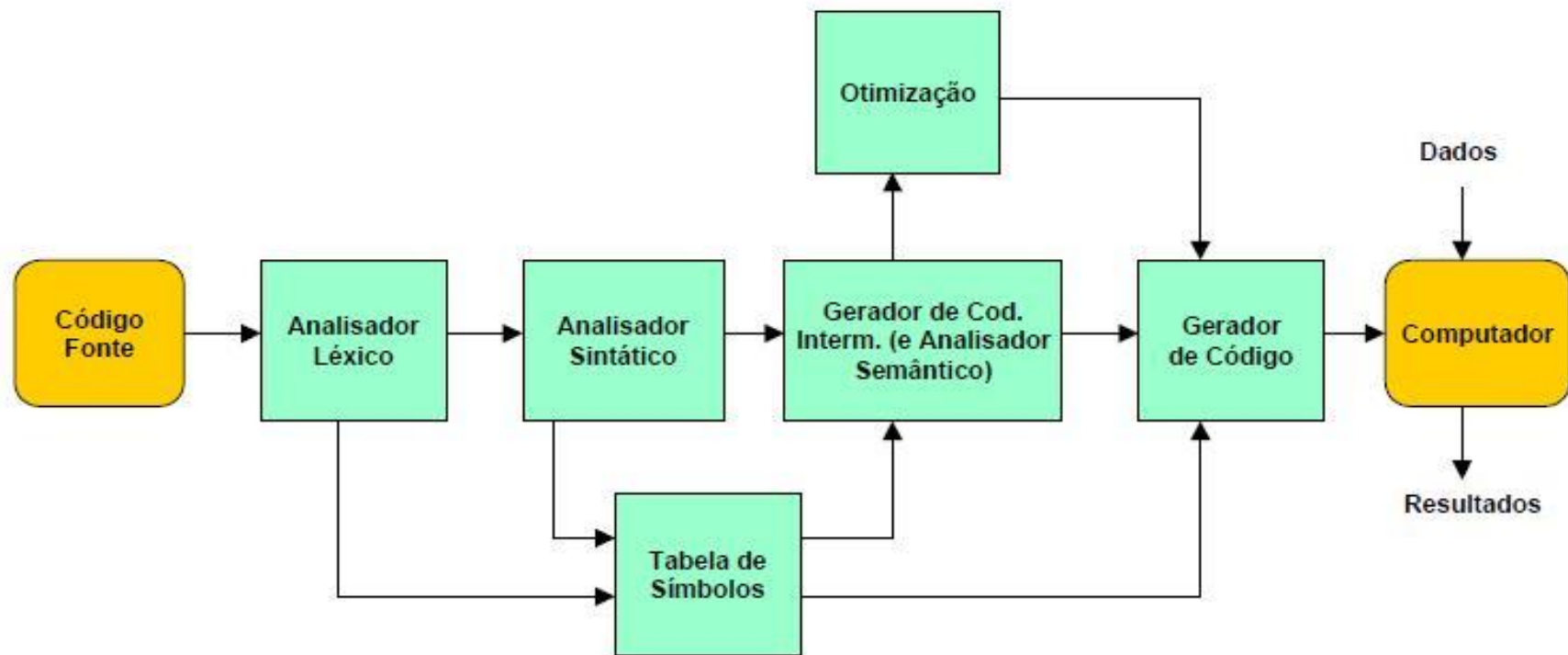
- Variáveis >> células de memória
- Atribuições >> operações
- Sequenciamento

## ◎ **Objetivos**

### ◎ Métodos de implementação

- **Compilação:** programa fonte é convertido para o nível de máquina para ser executado

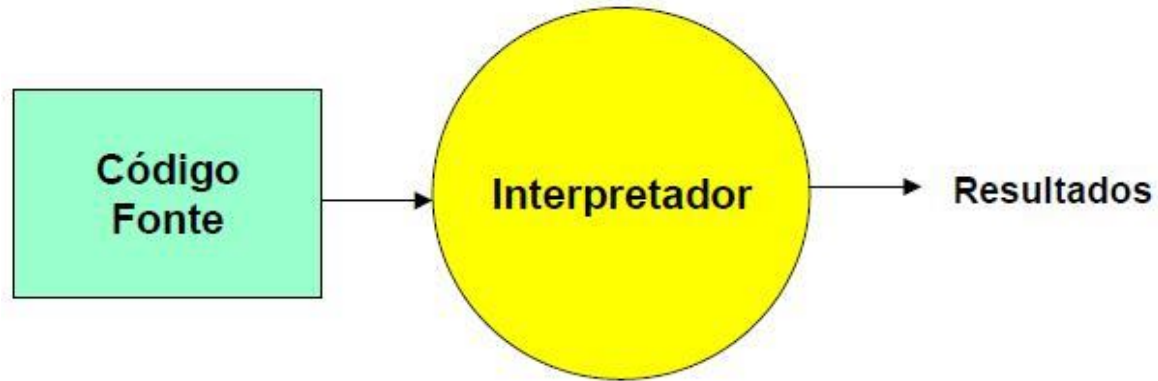




## ◎ **Objetivos**

### ◎ Métodos de implementação

- **Interpretação:** cada instrução é decodificada e executada imediatamente
- Simula máquina virtual
- *Debugging* mais simples
- Mais custoso (10 a 100 vezes + lenta que a compilação)



## Objetivos

### Métodos de implementação

- Híbrida



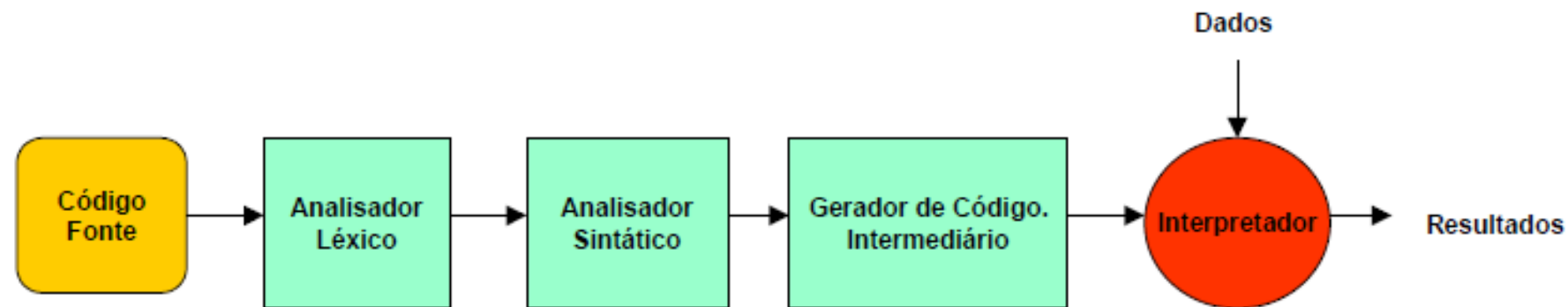
## ◎ Objetivos

### ◎ Métodos de implementação

#### ■ Implementação **híbrida**

- Compilador transforma o fonte em *bytecodes*
- *Bytecodes* são instruções compreendidas pela JVM
- A JVM é um interpretador que transforma as instruções em linguagem de máquina
- “*Write once, run anywhere*”
- Vantagem >> portabilidade!!!





## Objetivos

- ◉ Ambientes de programação



## ◎ **Objetivos**

### ◎ Enfim, os paradigmas...

- Imperativo
- Lógico
- Funcional
- Orientado a objetos

## ◎ Objetivos

### ◎ Enfim, os paradigmas...

#### ■ Imperativo

- *Imperare*: comandar
- Comandos atualizam variáveis armazenadas na memória
- Domínio: aplicações comerciais, SO's, etc.
- Exemplos: C, Pascal, ADA, Fortran, Lua, Cobol....

## ◎ Objetivos

### ◎ Enfim, os paradigmas...

#### ■ Lógico

- Implementa uma relação ao invés de um mapeamento
- Domínio: Sistemas especialistas (IA)
- Exemplos: PROLOG



## ◎ Objetivos

### ◎ Enfim, os paradigmas...

#### ■ Funcional

- Baseia-se em princípios matemáticos >> funções
- Iterativo (imperativo) vs. recursividade
- Domínio: matemática, IA, etc.
- Exemplos: LISP, ML, Schema, Haskell....

# ◎ Objetivos

## ◎ Enfim, os paradigmas...

### ■ **Orientado a objetos**

- Baseia-se no conceito de classes
- Atributos + métodos = classes
- Domínio: todos
- Exemplos: **JAVA**, C++, Simula 67....

## ◎ **Definição de linguagens de programação**

### ◎ O que é uma linguagem de programação?

“Qualquer notação para a descrição de algoritmos e estruturas de dados, embora, usualmente é requerido que uma linguagem de programação seja implementada em um computador”.

Pratt, Terrence

## ◎ Definição de linguagens de programação

**HTML é uma linguagem de programação?**

